Energy-Efficient Connected-Coverage in Wireless Sensor Networks

Ionut Cardei and Mihaela Cardei
Department of Computer Science and Engineering,
Florida Atlantic University,
777 Glades Road,
Boca Raton, FL 33431, USA
Fax: +1-561-297-2800
E-mail: icardei@cse.fau.edu
E-mail: mihaela@cse.fau.edu

Abstract: This paper addresses the target coverage problem in wireless sensor networks. Communication and sensing consume energy, therefore efficient power management can extend network lifetime. In this paper we consider a large number of sensors randomly deployed to monitor a number of targets.

Each target may be redundantly covered by multiple sensors. To conserve energy, we organize sensors in sets activated sucessively. In this paper we introduce the Connected Set Covers (CSC) problem that has as objective finding a maximum number of set covers such that each sensor node to be activated is connected to the base station. A sensor can participate in multiple sensor sets, but the total energy spent in all sets is constrained by the initial energy reserves. We show that the CSC problem is NP-complete and we propose three solutions: an integer programming based solution, a greedy approach, and a distributed and localized heuristic. Simulation results that validate our approaches are also presented.

Keywords: wireless sensor networks; energy efficiency; sensor scheduling; integer programming; optimization.

Biographical notes: Ionut Cardei is Assistant Professor at the Florida Atlantic University in Boca Raton, Florida, USA. He received his Ph.D. from the University of Minnesota in 2003. His main research interests are in wireless communication protocols and component-based software engineering.

Mihaela Cardei is Assistant Professor in the Department of Computer Science and Engineering at Florida Atlantic University, and Director of the NSF-funded Wireless and Sensor Network Laboratory. Dr. Cardei received her PhD and MS in Computer Science from the University of Minnesota, Twin Cities, in 2003 and 1999 respectively. Her research interests include wireless networking, wireless sensor networks, network protocol and algorithm design, and resource management in computer networks. She is a member of IEEE and ACM.

1 INTRODUCTION

Wireless sensor networks (WSNs) constitute the foundation of a broad range of applications related to national security, surveillance, military, health care, and environmental monitoring. One important class of WSNs is wireless ad-hoc sensor networks, characterized by an *ad-hoc* or *random* sensor deployment method, where the sensor location is not known a priori. This applies when individual sensor placement is infeasible, such as in battlefields or in disaster areas. Generally, more sensors are deployed than required (compared with the optimal placement) to perform the proposed task; this compensates for the lack of exact positioning and improves fault tolerance. The characteristics of a sensor network (Akyildiz et al. (2002)) include limited resources, large and dense networks, and a dynamic topology.

An important issue in sensor networks is power scarcity, driven in part by battery size and weight limitations. Mechanisms that optimize sensor energy utilization have a great impact on prolonging the network lifetime. Power saving techniques can generally be classified in two categories: scheduling the sensor nodes to alternate between active and sleep mode, and adjusting the transmission and/or sensing range of the wireless nodes. In this paper we deal with the first method. We design a scheduling mechanism in which only some of the sensors are active, while all other sensors are in sleep mode. This is the first paper to address the base station (BS) connectivity requirement in the target coverage problem. BS-connectivity requires that sensors in each sensor set are connected to the base station through a multihop path.

In this paper we address the *target coverage* problem. The goal is to maximize the network lifetime of a power constrained wireless sensor network deployed for monitoring a set of targets with known locations. We consider a large number of sensors deployed randomly in close proximity of a set of targets, that send the sensed information to a base station for processing. The method used to extend the network's lifetime is to organize the sensors into a number of sets, such that all the targets are monitored continuously. Additionally, energy constraints for each sensor and BS-connectivity of each sensor set must be satisfied. Besides reducing the energy consumed, this method lowers the density of active nodes, thus reducing interference at the MAC layer.

The contributions of this paper are: (1) introduce the Connected Set Covers (CSC) problem and prove its NP-completeness, (2) model the CSC problem using Integer Programming (IP) and design an IP-based heuristic (IP-CSC heuristic) that efficiently computes the sensor sets, (3) design a greedy-based heuristic (Greedy-CSC heuristic), (4) design a distributed and localized heuristic for the CSC problem, and (5) analyze the performance of these algorithms through simulations.

The rest of the paper is organized as follows. In section 2 we present related works on sensor coverage problems. Section 3 defines the CSC problem and shows its NP-completeness, section 4 presents the IP formulation and IP-based heuristic, section 5 describes the greedy mechanism, and section 6 presents the distributed and localized solution. In section 7, we present the simulation results and in section 8, we conclude the paper.

2 RELATED WORK

In this paper we address the sensor coverage problem. As pointed out by Meguerdichian et al. (2001), the coverage concept is a measure of the quality of service (QoS) of the sensing function and is subject to a wide range of interpretations due to a large variety of sensors and applications. The goal is to have each location in the physical space of interest within the sensing range of at least one sensor. A survey on coverage problems in wireless sensor networks is presented in Cardei and Wu (2006). The coverage problems can be classified in the following types: (1) area coverage (Carle and Simplot (2004); Tian and Georganas (2002); Wang et al. (2003); Zhang and Hou (2004)), where the objective is to cover an area, (2) point coverage (Cardei and Du (2005); Cardei et al. (2005); Cheng et al. (2005)), where the objective is to cover a set of targets, and (3) coverage problems that have the objective to determine the maximal support/breach path that traverses a sensor field (Meguerdichian et al. (2001)).

An important method for extending the network lifetime for the area coverage problem is to design a distributed and localized protocol that organizes the sensor nodes in sets. The network activity is organized in rounds, with sensors in the active set performing the area coverage, while all other sensors are in the sleep mode. Set formation is done based on the problem requirements, such as energy-efficiency, area monitoring, connectivity, etc. Different techniques have been proposed in the literature (Carle and Simplot (2004); Tian and Georganas (2002); Wang et al. (2003); Zhang and Hou (2004)) for determining the eligibility rule, that is, to select which sensors will be active in the next round.

For applications that require more stringent fault-tolerance or for positioning applications, k-coverage might be a requirement. In Huang and Tseng (2003), the goal is to determine whether a given area satisfies the k-coverage requirement, when each point in the area of interest is covered by at least k sensors. Both uniform and non-uniform sensing ranges are considered, and the k-coverage property is reduced to the k perimeter coverage of each sensor in the network.

A different coverage formulation is given by Meguerdichian et al. (2001). A path has the worst (best) coverage if it has the property that for any point on the path, the distance to the closest sensor is maximized (minimized). Given the initial and final locations of an agent, and a field instrumented with sensors, Meguerdichian et al. (2001) proposed centralized solutions to the worst (best) coverage based on the observation that worst coverage path lies on the Voronoi diagram lines and best coverage path lies on Delaunay triangulation lines.

The works most relevant to our approach are Cardei and Du (2005) and Cardei et al. (2005). The first article introduces the target coverage problem, where disjoint sensor sets are modeled as disjoint set covers, such that every cover completely monitors all the target points. The disjoint set coverage problem is proved to be NP-complete, and a lower bound of 2 for any polynomial-time approximation algorithm is indicated. The disjoint set cover problem described by Cardei and Du (2005) is reduced to a maximum flow problem, which is then modeled as mixed integer programming. This problem is further extended by Cardei et al. (2005) and Berman et al. (2004), where sensors are not restricted to participation in only disjoint sets, that is, a sensor can be active in more than one set. Berman et al. (2004) is the first article to propose an approximation algorithm for a point coverage problem. Still these works deal only with the coverage requirement, and do not address connectivity.

The coverage breach problem is introduced in

Cheng et al. (2005), addressing the case when sensor networks have limited bandwidth. The objective of the problem is to organize the sensors in disjoint sets, such that each set has a given bounded number of sensors and the overall breach is minimized. The overall breach is measured as the number of targets uncovered by the sensor sets.

Our paper is an extension of the maximum set covers problem addressed in Cardei et al. (2005), when an additional requirement, the BS-connectivity of each sensor set, is imposed. This is an important property for WSN, since sensor data has to be forwarded to the BS for further processing.

3 PROBLEM DEFINITION

Let us assume a homogeneous sensor network comprised of N sensors s_1, s_2,..., s_N randomly deployed to cover (monitor) M targets r_1, r_2,..., r_M. Each sensor has an initial energy E, communication range R_c, and sensing range R_s, with $R_s \leq R_c$. We assume a sensor node consumes e_1 energy per time unit for sensing and e_2 energy per time unit for communication.

We assume a base station, noted as the node s_0, is located in sensors' vicinity, within communication range of at least one sensor node. Once the sensors are deployed, they schedule their activity and communicate the sensed data to the BS. Again, the sensing objective is to continuously cover the M given targets.

One method to compute the sensor-target coverage relationship is to consider that a sensor covers a target if the Euclidean distance between the sensor and target is no greater than the sensing range. We also assume that two nodes can communicate if the Euclidean distance between them is less than or equal to the communication range R_c.

In order to conserve the energy resources and thus to prolong the network lifetime, we schedule the sensor nodes activity to alternate between sleep and active mode, such that all the targets are continuously monitored by the set of active sensors. The active set of sensors must be *BS-connected*, that means, for each active sensor s_u there must be a path of active sensors connecting s_u with the BS. BS-connectivity is an important requirement to assure that sensor data can be forwarded to the BS. The formal problem definition is given below:

Definition 1: **Target Connected-Coverage Problem**
Given M targets with known location, a base station, and an energy constrained WSN with N sensors randomly deployed in the targets' vicinity, schedule the sensor nodes' activity such that (1) all targets are continuously observed, (2) active sensors are BS-connected, and (3) network lifetime is maximized.

We measure the network lifetime as the time all M targets are monitored by a subset of BS-connected sensors in the active state, while satisfying the sensor energy constraint. The approach that we used in this paper for maximizing the network lifetime is to organize the sensors in set-covers. The network activity is organized in rounds, such that each set-cover is active in one round. Each round takes time δ, and only the sensors in the active set-cover are responsible for monitoring the targets and communication, while all other sensors are in sleep mode.

Next, we formally define the Connected Set Covers (CSC) problem that we use to solve the target connected-coverage problem.

Definition 2: **CSC Problem**
Given a set of sensors $s_1, s_2, \ldots, s_N$, a base station s_0, and a set of targets $r_1, r_2, \ldots, r_M$, find a family of set-covers $c_1, c_2, ..., c_K$ such that (1) K is maximized, (2) sensors in each set c_k ($k = 1, \ldots, K$) are BS-connected, (3) each sensor set monitors all targets, and (4) each sensor appearing in the sets $c_1, c_2, ..., c_K$ consumes at most E energy.

In CSC definition, the requirement to maximize K is equivalent with maximizing the network lifetime. Other requirements include targets' coverage by the active sensor set, active sensor set connectivity, and satisfying the sensor energy constraints. A sensor can participate in multiple sets and thus the sensor sets do not need to be disjoint. In Cardei et al. (2005), we introduced the Maximum Set Covers (MSC) problem, where we address only the requirements (1), (3), and (4). We proved that MSC is NP-complete. The CSC problem adds the BS-connectivity requirement to the MSC problem.

The CSC problem is NP-complete. We show that the MSC problem from (Cardei et al. (2005)) is a particular case of CSC problem: let us consider the CSC problem for the case when R_c is equal with the network diameter, $R_c = \max_{i,j=0\ldots N} dist(s_i, s_j)$. Therefore, any sensor can directly communicate with the BS, and any selected sensor set will be BS-connected. It follows that CSC reduces to the MSC problem for this case.

Since the MSC problem is NP-complete (Cardei et al. (2005)) and is a particular case of the CSC problem, and because CSC belongs to the NP-class, it follows that the CSC problem is NP-complete.

4 INTEGER PROGRAMMING SOLUTION

We first formulate the *CSC* problem using Integer Programming (IP) in subsection 4.1 and then propose an IP-based heuristic in subsection 4.2. This heuristic is centralized and is executed on the BS node. Once the sensors are deployed, they send their coordinates to the BS. The BS computes the connected set-covers and broadcasts back the sensors' schedule.

4.1 Integer Programming Formulation of the CSC Problem

In order to model the connectivity requirement for each set cover c_k, $k = 1, \ldots, K$, we represent the sensor network as a flow network $G' = (V', E')$. The vertex set V' contains

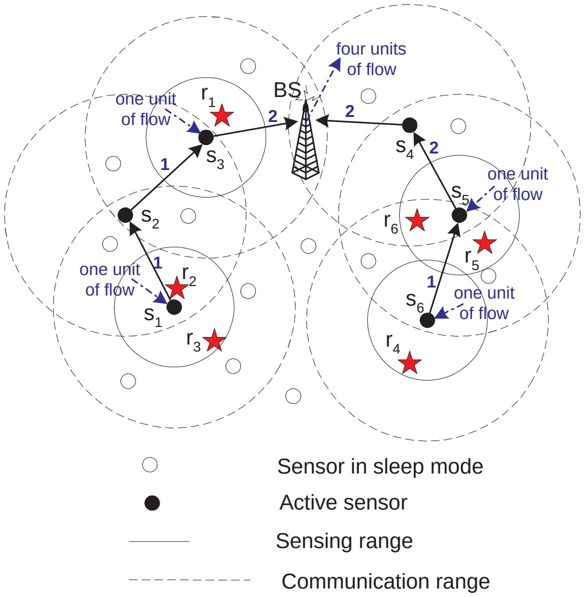

Figure 1: Example showing the gathering during a round

the sensor nodes and the BS, thus $V' = \{s_0, s_1, \ldots, s_N\}$. Note that in this paper we refer to a sensor s_u as the node (or vertex) u interchangeably. E' is the set of directed edges. An edge $(v, u) \in E'$ if node u is within the communication range of node v, that means $dist(s_v, s_u) \leq R_c$.

We assign a capacity $d_{vu} = M$ to each edge $(v, u) \in E'$. If $(v, u) \notin E'$ then $d_{vu} = 0$. Since we assumed a connected wireless network, the graph G' is strongly connected as well. We note with f_{vuk} the flow sent from node v to node u in the set cover c_k, where $f : V' \times V' \times \{1, 2, \ldots, K\} \to R$ and $f_{vuk} \geq 0$.

The flow values and the source vertices vary with each set-cover c_k. For a set-cover c_k, the nodes performing the sensing task are the sources in the flow network, and the BS is the sink. One unit of flow is inserted by each source vertex.

Figure 1 shows an example of a flow network during a round. In this round, there are six active sensors: four sensing nodes s_1, s_3, s_5 and s_6, and two relay nodes s_2 and s_4. The four sensing nodes cover all six targets $r_1, \ldots, r_6$, while the two relay nodes help achieving BS-connectivity. Therefore, in this round there are four sources where one unit of flow is inserted, and one sink that receives four units of flow. If each round is active time δ, the sensing and communication energy is computed as $E_1 = e_1\delta$, and $E_2 = e_2\delta$. Then the total energy spent in this round by all nodes is $4E_1 + 6E_2$.

IP formulation
Given:

- N sensor nodes $s_1, \ldots, s_N$ and a Base Station s_0.

- M targets r_1, r_2,..., r_M.

- E, the initial energy of each sensor, and δ the time each set-cover is active; the sensing energy consumed per round by an active node performing the sensing task is $E_1 = e_1\delta$, and the communication energy consumed per round by an active node is $E_2 = e_2\delta$.

- the coefficients showing the coverage relationship between sensors and targets: $a_{uj} = 1$ if sensor s_u covers the target r_j, that means $dist(s_u, r_j) \leq R_s$. Otherwise, $a_{uj} = 0$.

- d_{vu}, the capacity of an edge (v, u) in the flow network G'. If $(v, u) \in E'$, then $d_{vu} = M$, otherwise $d_{vu} = 0$.

Variables:

- c_k, boolean variable, for $k = 1, \ldots, K$; $c_k = 1$ if this subset is a valid connected set-cover, otherwise $c_k = 0$.

- x_{uk1} and x_{uk2}, boolean variable, for $u = 1, \ldots, N$, and $k = 1, \ldots, K$;
  $x_{uk1} = 1$ if sensor u is sensing in the set cover c_k, otherwise $x_{uk1} = 0$.
  $x_{uk2} = 1$ if sensor u is communicating in the set cover c_k, otherwise $x_{uk2} = 0$.
  There are three valid cases for a sensor u and a set cover c_k: (1) sensor u is in the sleep mode, and thus $x_{uk1} = x_{uk2} = 0$, (2) sensor u is a *sensing node*, that means node u is sensing and communicating, and thus $x_{uk1} = x_{uk2} = 1$, and (3) sensor u is a *relay node*, active only for communication (relaying), and thus $x_{uk1} = 0$ and $x_{uk2} = 1$.

- $f_{vuk} \in [0, M]$, integer variable, for any edge $(v, u) \in E'$ and for any $k = 1..K$.

Objective:
Maximize $c_1 + c_2 + \cdots + c_K$

Subject to the following constraints:

- $\sum_{k=1}^{K} (x_{uk1}E_1 + x_{uk2}E_2) \leq E$, for all $u = 1, \ldots, N$ (1)

- $\sum_{u=1}^{N} x_{uk1}a_{uj} \geq c_k$ for all $j = 1, \ldots, M; k = 1, \ldots, K$ (2)

- $c_k \geq x_{uk2}$ for all $u = 1, \ldots, N; k = 1, \ldots, K$ (3)

- $x_{uk2} \geq x_{uk1}$ for all $u = 1, \ldots, N; k = 1, \ldots, K$ (4)

- $\sum_{v=1}^{N} f_{vuk} - \sum_{w=0}^{N} f_{uwk} = x_{uk1}$ for all $u = 1, \ldots, N; k = 1, \ldots, K$ (5)

- $\sum_{v=1}^{N} f_{v0k} = \sum_{u=1}^{N} x_{uk1}$ for all $k = 1, \ldots, K$ (6)

- $f_{uvk} \leq d_{uv}$ for all $u = 1, \ldots, N$; $v = 0, \ldots, N$; $k = 1, \ldots, K$ (7)

- $f_{uvk} \leq \sum_{i=1}^{N} x_{ik1}$ for all $u = 1, \ldots, N; v = 0, \ldots, N$; $k = 1, \ldots, K$ (8)

- $\sum_{v=1}^{N} f_{vuk} \leq x_{uk2}M$ for all $u = 1, \ldots, N; k = 1, \ldots, K$ (9)

where:

- $c_k, x_{uk1}, x_{uk2} \in \{0, 1\}$, for $u = 1, \ldots, N; k = 1, \ldots, K$
- $f_{uvk}, f_{v0k} \in \{0, 1, 2, \ldots, M\}$, for $u, v = 1, \ldots N; k = 1, \ldots, K$

Remarks:

1. Constraint (1) guarantees that the total energy consumed by each sensor s_u, $u = 1, \ldots, N$ is less than or equal to E, which is the initial energy for each sensor.
2. The second constraint (2) guarantees that for each valid set cover c_k (that is $c_k = 1$), each target is covered by at least one sensing node.
3. Constraint (3) assures that if a set-cover is invalid ($c_k = 0$) then no sensor participates in sensing or communication.
4. Constraint (4) assures that if a sensor s_u is sensing in the set-cover c_k ($x_{uk1} = 1$), then it has to communicate as well (and thus $x_{uk2} = 1$) in order to be able to forward its sensed data.
5. Constraint (5) verifies the flow conservation property. If a sensor s_u is not a sensing node in the set-cover c_k, then the flow in equals the flow out. If s_u is a sensing node in the set-cover c_k, then s_u is a source, and thus an additional one unit flow is inserted.
6. Constraint (6) represents the flow conservation at the BS (node s_0). BS is the sink of the flow network, and thus it must receive $\sum_{i=1}^{N} x_{ik1}$ units of flow.
7. Constraint (7) is the capacity constraint property.
8. Constraint (8) assures that the flow of any edge is at most $\sum_{i=1}^{N} x_{ik1}$ units, which is how many flow units have been inserted through the sources of the flow network.
9. Constraint (9) refers to case when a sensor s_u is in the sleep mode during the round c_k ($x_{uk2} = 0$). In this case, the flow in at node s_u is zero. Based on condition (4), it results that the flow out at node s_u is zero as well.
10. K represents an upper bound for the number of covers.

4.2 Integer Programming-based Heuristic

In this subsection, we propose a solution for the CSC problem, using the Integer Programming (IP) formulation in subsection 4.1. Based on the solution returned by the IP solver ($\bar{x}_{uk1}$, $\bar{x}_{uk2}$, $\bar{f}_{vuk}$, and $\bar{c}_k$), our heuristic constructs the connected set-covers. For each set cover $\bar{c}_k$, we check if this is a valid set cover, that is if $\bar{c}_k = 1$. If the equality is true, then a new set cover (S_h, R_h) is formed. The set S_h (set R_h) contains the sensing sensors (relay sensors respectively) in the set-cover h. All other sensors are in sleep mode.

The complexity of our heuristic is dominated by the complexity of the Integer Programming solver. The network lifetime is computed as $\delta \sum_{k=0}^{K} c_k$. Simulations results are presented in section 7.

IP-CSC

solve the IP from subsection 4.1 and get the solution $\bar{x}_{uk1}$, x_{uk2}, f_{vuk}, and c_k
$h \leftarrow 0$
for all $i \leftarrow 1$ to K **do**
  if $c_i = 1$ **then**
    $h \leftarrow h + 1$; $S_h \leftarrow \phi$; $R_h \leftarrow \phi$;
    for all $j \leftarrow 1$ to N **do**
      if $x_{ji1} = 1$ **then**
        $S_h \leftarrow S_h \cup \{s_j\}$
      end if
      if $x_{ji2} = 1$ && $x_{ji1} \neq 1$ **then**
        $R_h \leftarrow R_h \cup \{s_j\}$
      end if
    end for
  end if
end for
return the connected set covers (S_1, R_1), (S_2, R_2), … (S_h, R_h).

5 A GREEDY HEURISTIC FOR THE CSC PROBLEM

The heuristic *Greedy-CSC* is executed by the base station. Once the sensors are deployed, they send their coordinates to the BS. The BS computes the connected set-covers and broadcasts back the sensors' schedule.

Greedy-CSC

set residual energy of each sensor u to $E'_u \leftarrow E$
$SENSORS \leftarrow \{(s_1, E'_1), \ldots, (s_N, E'_N)\}$
$h \leftarrow 0$
while the sensors in $SENSORS$ are connected and cover all the targets **do**
  /* a new set C_h will be formed */
  $h \leftarrow h + 1$; $S_h \leftarrow \phi$; $R_h \leftarrow \phi$
  $TARGETS \leftarrow \{r_1, r_2, \ldots, r_M\}$
  while $TARGETS \neq 0$ **do**
    /* more targets have to be covered */
    find a critical target $r_{critical} \in TARGETS$
    select a sensor $s_u \in SENSORS$ with greatest contribution that covers $r_{critical}$ and has $E_u \geq E_1 + E_2$
    $S_h \leftarrow S_h \cup \{s_u\}$
    for all targets $r_j \in TARGETS$ **do**
      if r_j is covered by s_u **then**
        $TARGETS \leftarrow TARGETS - r_j$
      end if
    end for
  end while

19: apply Breadth-First Search (BFS) algorithm starting from the BS
20: prune edges in the BFS tree such that to maintain paths from the BS to each sensor in S_h
21: **for all** sensors s_u in the resulted BFS subtree with $s_u \notin S_h$ **do**
22: $R_h \leftarrow R_h \cup \{s_u\}$
23: **end for**
24: **for all** sensors $s_u \in S_h \cup R_h$ **do**
25: **if** $s_u \in S_h$ **then**
26: $E'_u \leftarrow E'_u - (E_1 + E_2)$
27: **end if**
28: **if** $s_u \in R_h$ **then**
29: $E'_u \leftarrow E'_u - E_2$
30: **end if**
31: **if** $E'_u < E_2$ **then**
32: $SENSORS \leftarrow SENSORS - s_u$
33: **end if**
34: **end for**
35: **end while**
36: return h-number of connected set covers and the connected set-covers (S_1, R_1), (S_2, R_2), $\dots$, (S_h, R_h).

The heuristic recursively builds the set covers, in lines 4 to 34. The residual energy of each sensor is set initially to E. Each connected set-cover corresponds to a round that will be active for δ, and, thus, an active sensing sensor consumes $E_1 + E_2$ energy, while a relay sensor consumes E_2 energy per round. The set *SENSORS* maintains the set of sensors that have at least E_2 energy left, and can therefore participate in additional set-covers.

Similar with IP-CSC heuristic, we note with S_h (R_h) the set containing the sensing sensors (relay sensors respectively) in the current set-cover h. We first construct the sensing sensor set S_h, in lines 8 to 18 and then the relay sensor set R_h in lines 19 to 23. The set *TARGETS* contains the targets that still have to be covered by the current set h. At each step, a critical target is selected, in line 10, to be covered. This can be for example the target most sparsely covered, both in terms of number of sensors as well as with regard to the residual energy of those sensors. The sensors considered for sensing have to be in the set *SENSORS* and to have at least E_1+E_2 residual energy. Once the critical target has been selected, the heuristic selects the sensor with the greatest contribution that covers the critical target. Various sensor contribution functions can be defined. For example we can consider a sensor to have greater contribution if it covers a larger number of uncovered targets and if it has more residual energy available. Once a sensor has been selected, it is added to the current set cover in line 12, and all additionally covered targets are removed from the $TARGETS$ set.

After all the targets have been covered, we need to guarantee BS-connectivity. For this we run the Breadth-First Search (BFS) algorithm (Cormen (2001)) starting from the BS. We prune the resultant BFS tree and keep only the subtree that connects the BS with the nodes in S_h. This can be done starting from the nodes in S_h and following the predecessor values, until we reach the BS. The non-sensing nodes in the resultant subtree are the relay nodes, and are therefore included in R_h.

After a set cover (S_h, R_h) has been formed, the residual energy each sensor in $S_h \cup R_h$ is updated in lines 25 to 30. If, after the update, the residual energy of a sensor is less that E_2, then that sensor is removed from the set *SENSORS*. This is because that sensor cannot participate as a sensing or relay sensor in another set-cover.

The network lifetime is computed as $h\delta$, with h the number of set covers and δ the time each cover is active. The complexity of the Greedy-CSC Heuristic is $O(h \cdot (V' + E' + M^2N^2))$, where the graph $G' = (V', E')$ has been defined in the subsection 4.1. The variable h is upper-bounded by $d \cdot E/(E_1 + E_2)$, where d is the number of sensors that cover the most sparsely covered target. The complexity of the BFS is $O(V' + E')$ (Cormen (2001)). As the energy values E_1, E_2 and E are constant, the heuristic runtime is $O(d \cdot (V' + E' + M^2N^2))$.

6 DISTRIBUTED AND LOCALIZED HEURISTIC

In this section we propose a distributed and localized heuristic. By *distributed and localized* we refer to a decision process at each node that makes use of only information for a neighborhood within a constant number of hops. A distributed and localized approach is desirable in wireless sensor networks since it is scalable, and adapts better to dynamic and large topologies.

The distributed algorithm runs in rounds. Each round begins with an initialization phase, where sensors will decide whether they will be in active mode (sensing or relaying) or sleep mode during the current round. The initialization time W should be far less than the duration of a round. Each initialization phase has two steps, **sensing node selection** and **relay node selection**. In the first step, the sensing sensors are determined, and in the second step the relay sensors are decided, such that to ensure BS-connectivity. Let us consider that the first step takes W_1 time and the second step takes W_2 time, then the initialization phase takes $W = W_1 + W_2$.

Similar with our previous solutions, we assume that an active sensor consumes E_1 energy for sensing and E_2 energy for communication per round, starting from an initial energy E. We assume sensor nodes are synchronized, and we measure the network lifetime as the number of rounds computed.

Next, we describe in more detail each of the two steps of the initialization phase.

6.1 Sensing Nodes Selection

In this section we describe how a sensor s_u decides whether or not it will be a *sensing node* during the current round. Let us consider the following notations:

- E'_u is the residual energy of s_u
- E is the initial energy
- the set M_u contains all the targets located within the sensing range of s_u
- the set $TARGETS_u$ is maintained by s_u and contains all the targets in s_u's sensing range that are not covered by any node that has declared and advertised as a *sensing node* until now. This set is updated continuously by s_u based on broadcast messages received from 2-hop neighbor sensing nodes.
- M is the number of targets

Sensor node s_u computes a backoff time $T_u \leq W_1$. If s_u cannot become a sensing node due to the energy constraints ($E'_u < E_1 + E_2$), then $T_u = W_1$.

Otherwise, T_u is computed as $T_u = (1 - \alpha\frac{E'_u}{E} - \beta\frac{|TARGETS_u|}{M}) \cdot W_1$, where α and β are parameters used to decide the weight of residual energy versus the weight of the number of uncovered targets in computing the backoff time, $\alpha + \beta < 1$. Parameters α and β are initialized at the beginning of the application and do not change during the application lifetime. The rationale of this formula is to give higher priority (smaller T_u) to sensors that have higher residual energy and cover a larger number of uncovered targets.

When T_u expires, if $TARGETS_u \neq \emptyset$ and $E'_u \geq E_1 + E_2$, then s_u declares itself as a *sensing node* during the current round. Additionally, s_u broadcasts this decision together with the set M_u to its 2-hop neighbors. We use a 2-hop neighborhood since our goal is to advertise s_u's status to all other sensors that cover common targets with s_u. If two sensors cover a common target, then the distance between these sensors is at most $2R_s$, where $R_s \leq R_c$.

When a node s_v receives such an advertisement message, it updates its $TARGETS_v$ set and T_v timer accordingly. On the other hand, if $TARGETS_u$ becomes empty, then s_u will not be a sensing node in this round.

As different sensors have different waiting times, this serializes the sensors' broadcasts in their local neighborhood and gives priority to the sensors with higher residual energy that cover more uncovered targets. Since the 2-hops advertisement messages are very small, we neglect the energy consumed in forwarding them.

If $TARGETS_u \neq \emptyset$ when T_u expires and if $E'_u < E_1 + E_2$, then there are targets that cannot be covered in the current round and s_u sends this failure notification to the BS.

Next, we present the **Sensing Nodes Setup** procedure that is run by each sensor s_u, $u = 1, \ldots, N$:

Sensing Nodes Setup (s_u, α, β)

initialize the set M_u and set $TARGETS_u \leftarrow M_u$
if $E'_u \geq E_1 + E_2$ **then**
  compute waiting time $T_u \leftarrow (1 - \alpha\frac{E'_u}{E} - \beta\frac{|TARGETS_u|}{M})W_1$, and start timer t
else
  $T_u \leftarrow W_1$, and start timer t
end if
while $t \leq T_u$ and $TARGETS_u \neq \emptyset$ **do**
  if message from neighbor sensor is received **then**
    update $TARGETS_u$, by removing the targets now covered by the advertising sensing node; update the backoff timer T_u
    if $TARGETS_u = 0$ **then**
      return;
    end if
    update the waiting time T_u
  end if
end while
if $E'_u < E_1 + E_2$ **then**
  s_u reports failure to the BS, indicating the targets it cannot cover due to energy constraints
end if
s_u will be a *sensing node* in this round; s_u broadcast to its 2-hop neighbors its status and the set M_u
return

6.2 Relay Nodes Selection

The goal of this step is to assure BS-connectivity for the sensing sensor set selected in the first step. We assume that each sensor knows the location of all targets and the BS.

Each sensing node s_u builds a *virtual tree* T, where the set of vertices is comprised of the BS and the set of targets. Node s_u uses Prim's algorithm (Cormen (2001)) to construct a MST (minimum spanning tree) rooted at the BS. Since each sensing node has the same BS and target set information, each sensing node will build the same virtual tree T.

In order to uniquely associate each target with a sensor, we introduce the concept of *supervisor*. In the first step, when a node s_u decides to become a sensing node, it becomes the supervisor of the targets in the set $TARGETS_u$. That means s_u is the first sensing node that decides to monitor the targets in $TARGETS_u$, during this round. Note that once the initialization phase ends, s_u will actually monitor all the targets in M_u.

The sensor s_u considers all the targets it supervises, in turn. For each such target r_i, the node s_u initiates the selection of relay nodes along the *virtual link* $(r_i, \pi(r_i))$, where $\pi(r_i)$ is the predecessor (or parent) of the target r_i in the virtual tree T. The goal is to select relay nodes on the path between $s_u = supervisor(r_i)$ and the supervisor of $\pi(r_i)$, noted with $supervisor(\pi(r_i))$.

We know that $dist(supervisor(r_i), supervisor(\pi(r_i))) \leq dist(r_i, \pi(r_i)) + 2R_s$, where R_s is the sensing range of a sensor. Node s_u broadcasts a special control message $RELAY\_REQ$ containing the following information: s_u's location, the search distance $dist(r_i, \pi(r_i)) + 2R_s$, and the destination target (in our case $\pi(r_i)$). We execute a localized broadcasting, where only the nodes s_v with properties: (1) $dist(s_u, s_v) \leq search\ distance$, and (2) $E'_v \geq E_2$ participate in the data forwarding. The second property assures that a node has sufficient residual energy to eventually serve as a relay node.

Each node that receives and forwards *RELAY_REQ*, sets up a gradient toward the node from which it received the request. Such a gradient is maintained with a small timeout interval, sufficient to allow *RELAY_REP* message to propagate back to s_u. Various criteria can be considered to set-up the gradients. For example the gradient can be set toward the node from which the first *RELAY_REQ* was received, toward the node closer to s_u, or toward a node with higher residual energy.

When the node $supervisor(\pi(r_i))$ (our intended destination) receives the *RELAY_REQ* message, it will reply back to s_u with a *RELAY_REP* message. This message propagates back to the node s_u, along the gradients, and all the nodes that participate in this *RELAY_REP* forwarding become *relay nodes* during the current round.

To maintain the uniformity of this mechanism, we see the BS as both a target, as well as its own supervisor. Also, if a sensor is the supervisor of two targets and the two targets form a virtual link in T, then no relay node has to be selected for this virtual link. The time W_2 allocated for the relay nodes selection step is computed as an estimation of the round-trip time along the longest edge in the virtual tree T.

Next, we present the **Relay Nodes Setup** procedure that is run by each sensing node s_u:

Relay Nodes Setup (s_u)

build the virtual tree T, and set up the predecessors of each vertex in T
for each target r_i supervised by s_u **do**
  compute $search\ dist \leftarrow dist(r_i, \pi(r_i)) + 2R_s$
  broadcast $RELAY\_REQ(s_u, s_u$ *location*, *search dist*, $\pi(r_i))$; forwarding nodes set-up gradients, and $\pi(r_i)$ sends *RELAY_REP* along these gradients; all nodes that forward *RELAY_REP* become *relay nodes*.
  if timeout without receiving *RELAY_REP* **then**
    s_u sends failure notification to the BS
    return
  end if
end for
return

Theorem: The sensing and the relay nodes selected in the initialization phase guarantees BS-connectivity.

Proof: Let us take any sensing node s_u. We show that there is a path of active nodes (sensing or relay nodes) between s_u and the BS. Let us take any target r_i for which s_u is the supervisor. Let us note with $r_i, r_{i+1}, \ldots, r_j = BS$ the path between r_i and the BS, in the virtual tree T.

For any two consecutive nodes r_k and r_{k+1} on this virtual path, there is a path of active sensor nodes between the $supervisor(r_k)$ and $supervisor(r_{k+1})$, which was initiated by the $supervisor(r_k)$ using a *RELAY_INIT*, *RELAY_REP* message exchange mechanism. □

7 SIMULATION RESULTS

In this section we evaluate the performance of the following algorithms: *IP-CSC* from section 4.2, *Greedy-CSC* from section 5, and the distributed and localized algorithm presented in the section 6 and referred here as *Distr-CSC*. We simulate a stationary network with sensor nodes and target points located randomly in a 500m ×500m area. Additionally, we consider the following parameters:

- initial battery energy of each sensor is 1000mWh. The power used for sensing is $E_1 = 20$mW and the power necessary for communication (and processing) is $E_2 =$ 60mW.
- the sensing round duration is fixed at $\delta = 1$ hour.
- for *Distr-CSC*, parameters $\alpha = \beta = 0.4$.

We assume sensor nodes are homogeneous, meaning all sensors have the same sensing range and the same communication range for a specific scenario.

The main performance metric we focus on is the number of covers determined by the *CSC* algorithms, which is the number of successive rounds full target coverage is guaranteed. We use this parameter as an indicator for the network lifetime. Each cover in the cover set is verified for correctness, checking whether 1) all targets are within sensing range of at least a sensor from the cover, and 2) all sensors from the cover can reach the base station using only other relay sensors from the same set.

In the simulation we consider the following tunable parameters:

- N, the number of sensor nodes. We vary the number of randomly deployed sensor nodes between 20 and 600 to study the effect of node density on performance.
- M, the number of targets to be covered. We vary the number of targets between 10 and 100.
- R_c, the communication range. We vary the communication range between 80m and 200m.

We have implemented the *IP-CSC* solution using the ILOG CPLEX optimization library. For *Distr-CSC*, we have implemented a custom event-based simulator in C++. We assumed the energy expended during phases W_1 and W_2 to be negligeable compared to the energy spent during a sensing round δ. We also assume reliable communication between neighbor nodes. As part of our future work we will enhance the protocol to cope with a non-ideal communication channel.

In the first set of experiments we compare *Greedy-CSC* with *Distr-CSC* in larger networks. *IP-CSC* has a very high computation complexity and runtime is prohibitively

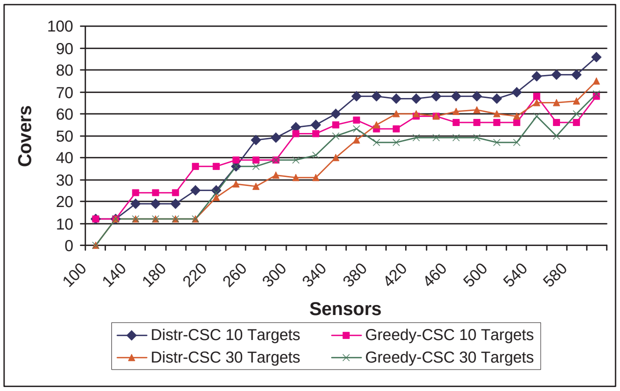

Figure 2: Number of covers obtained by Greedy-CSC and Distr-CSC with variable number of sensors

long. We compare *IP-CSC* with the other two heuristic algorithms in the second part of this section.

In the first experiment, illustrated in Figure 2, we vary the number of sensors from 100 to 600 and we measure the cover set size for networks with 10 and 30 targets. The sensing range is set to 50m and the communication range is set to 80m. As expected, scenarios with higher sensor density yield more covers. A scenario with more targets also requires larger sensing covers, thus reducing the overall number of covers.

We notice that, in general, the localized and distributed solution provides more covers than *Greedy-CSC*. This is because in *Distr-CSC* the *virtual tree*-based approach for finding relay nodes for R_h promotes reuse of already active supervisor nodes. On the other hand, *Greedy-CSC* forms a breadh-first-search (BFS) tree that shortens sensor - BS paths at the expense of increasing the number of relay nodes involved.

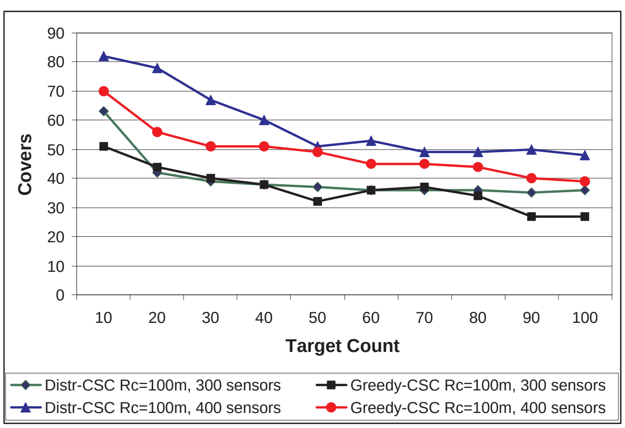

Figure 3: Number of covers obtained by Greedy-CSC and Distr-CSC with variable number of targets

In Figure 3, we present results from scenarios where we vary the number of targets between 10 and 100 and we run *Greedy-CSC* and *Distr-CSC* with 300 and 400 sensors, respectively. The communication range was set to 100m and the sensing range to 50m. By increasing the target count, more sensors may be required to be active at a time to guarantee coverage (set S_h). This implies that more sensor nodes will be assigned as relay nodes (set R_h), thus reducing the overall number of covers. We notice again that *Distr-CSC* performs better than *Greedy-CSC* for the same reasons described above.

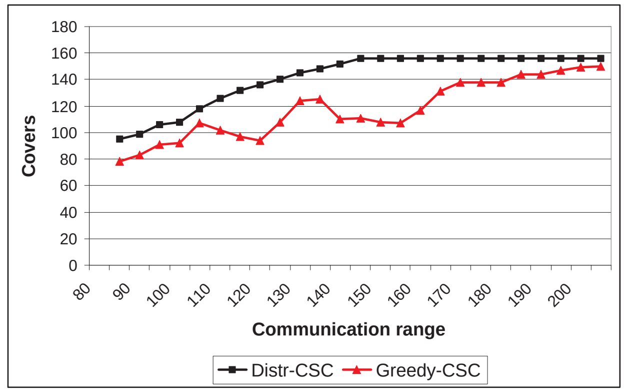

Figure 4: Number of covers obtained by Greedy-CSC and Distr-CSC with variable communication range

Figure 4 illustrates the number of covers in an experiment with 20 targets and 500 sensors, where the sensing range is 60m and the communication range varies from 80m to 200m. We notice that the number of covers increases with the communication range since there wil be fewer sensors involved in relaying. As R_c increases, the number of covers converges for both solutions because higher connectivity reduces the need for relay nodes. *Distr-CSC* reaches the maximum at $R_c = 150$m, after which communication range ceases to be a limiting factor.

In the second part of our simulations we implemented the integer programming presented in the section 4.1 using the ILOG CPLEX optimizer. We compared the *IP-CSC* solution with *Distr-CSC* and *Greedy-CSC* in smaller topologies, with 20 to 50 nodes and 5 targets. The initial energy reserve for a sensor node was set to 300mWh, the communication range to 16m and the sensing range to 12m.

Figure 5 shows the number of covers for the three algorithms with varying sensor count. *IP-CSC* obtains the highest number of covers while the other two algorithms get close results. Still, the running time of the integer programming solution is very high and thus it is not a practical approach for large topologies.

The simulation results can be summarized as follows:

- for a specific number of targets, the network lifetime output by our heuristics increases with the number of sensors and the communication range
- for a specific number of sensors and sensing range, the

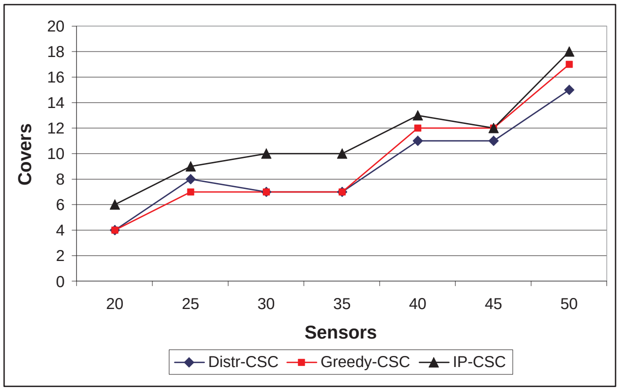

Figure 5: Number of covers obtained by IP-CSC, Greedy-CSC, and Distr-CSC with variable number of sensors

network lifetime decreases as the number of targets to be monitored increases

- in general, *Distr-CSC* performs better than *Greedy-CSC* because *Distr-CSC* employs a *virtual tree*-based approach for finding relay nodes that promotes reuse of already active supervisor nodes, while *Greedy-CSC* uses a BFS tree that results in shorter sensor - BS paths, involving more relay nodes. Since *Distr-CSC* is distributed and localized, it scales well with large sensor network topologies

- *IP-CSC* has the best performance as it is expected to find the optimal solution, but it has a long runtime and therefore is not feasible for large scenarios.

8 CONCLUSIONS

In this paper we addressed the connected set covers (CSC) problem, used to solve the target connected-coverage problem. The CSC problem addressed in this paper has as objective to determine maximum network lifetime when all targets are covered, sensor energy resources are constrained, and active sensors are BS-connected.

In this paper we showed CSC is NP-complete, and proposed three solutions for the CSC problem. The first two solutions are centralized, one using integer programming and the second using a greedy technique. The third approach is distributed and localized being suitable for practical implementation in sensor networks. We verified our approaches through simulation. Our future work is to test our approaches on different data gathering patterns, for both periodic and event-based data gathering.

ACKNOWLEDGMENT

This work is supported in part by NSF grants CNS 0422762 and CCF 0545488.

REFERENCES

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y. and Cayirci, E. (2002) 'A Survey on Sensor Networks', *IEEE Communications Magazine*, pp. 102-114, August.

Berman, P., Calinescu, G., Shah, C. and Zelikovsky, A. (2004) 'Power Efficient Monitoring Management in Sensor Networks', *IEEE Wireless Communications and Networking Conference.*

Cardei, M. and Du, D.-Z. (2005) 'Improving Wireless Sensor Network Lifetime through Power Aware Organization', *ACM Wireless Networks*, Vol. 11, No. 3, May.

Cardei, M., Thai, M., Li, Y. and Wu, W. (2005), 'Energy-Efficient Target Coverage in Wireless Sensor Networks', *IEEE INFOCOM'05*, March.

Cardei, M. and Wu, J. (2006), 'Energy-Efficient Coverage Problems in Wireless Ad Hoc Sensor Networks', *Computer Communications*, Vol. 29, No. 4, pp. 413-420, February.

Carle, J. and Simplot, D. (2004), 'Energy Efficient Area Monitoring by Sensor Networks', *IEEE Computer*, Vol. 37, No. 2, pp. 40-46.

Cheng, M., Ruan, L. and Wu, W. (2005), 'Achieving Minimum Coverage Breach under Bandwidth Constraints in Wireless Sensor Networks', *IEEE INFOCOM'05*, March.

Cormen, T. H., Leiserson, C. E., Rivest, R. L. and Stein, C. (2001), 'Introduction to Algorithms, 2nd edition, McGraw Hill.

CPLEX solver. Available at http://www.cplex.com.

Huang, C.-F. and Tseng, Y.-C. (2003) 'The Coverage Problem in a Wireless Sensor Network', *ACM MobiCom'03*, pp. 115-121, September.

Meguerdichian, S., Koushanfar, F., Potkonjak, M. and Srivastava, M. (2001) 'Coverage Problems in Wireless Ad-Hoc Sensor Networks',*IEEE INFOCOM'01*, pp. 1380-1387.

Tian, D. and Georganas, N. D. (2002) 'A Coverage-Preserving Node Scheduling Scheme for Large Wireless Sensor Networks', *ACM Workshop on Wireless Sensor Networks and Applications.*

Wang, X., Xing, G., Zhang, Y., Lu, C., Pless, R. and Gill, C. D. (2003) 'Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks', *First ACM Conference on Embedded Networked Sensor Systems.*

Zhang, H. and Hou, J. C. (2004) 'Maintaining Sensing Coverage and Connectivity in Large Sensor Networks', *NSF International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless and Peer-to-Peer Networks*, February.